A HighSpeed TCP Study: Characteristics and Deployment Issues

**** Paper Submitted to SC2003 ****

Evandro de Souza [*] [†]
EDSouza@lbl.gov
[*]Lawrence Berkeley National Lab, Berkeley, CA, USA
[†]State University of Campinas, Campinas, SP, Brazil

Deb Agarwal[‡]
DAAgarwal@lbl.gov
[‡]Lawrence Berkeley National Lab, Berkeley, CA, USA

***Abstract*— The current congestion control mechanism used in TCP has difficulty reaching full utilization on high speed links, particularly on wide-area connections. For example, the packet drop rate needed to fill a Gigabit pipe using the present TCP protocol is below the currently achievable fiber optic error rates. HighSpeed TCP was recently proposed as a modification of TCP's congestion control mechanism to allow it to achieve reasonable performance in high speed wide-area links. In this paper, simulation results showing the performance of HighSpeed TCP and the impact of its use on the present implementation of TCP are presented. Network conditions including different degrees of congestion, different levels of loss rate, different degrees of bursty traffic and two distinct router queue management policies were simulated. The performance and fairness of HighSpeed TCP were compared to the existing TCP and solutions for bulk-data transfer using parallel streams.**

I. Introduction

Applications demanding high bandwidth such as bulk-data transfer, multimedia web streaming, and computational grids are becoming increasingly prevalent in high performance computing. These applications are often operating over wide-area networks so, performance over the wide-area network has become a critical issue [1].

In the Internet, TCP (Transmission Control Protocol) has been widely used as a transport protocol. Many applications such as HTTP (Hyper Text Transfer Protocol) for the World Wide Web and FTP (File Transfer Protocol) are based on TCP. Recent experience indicates that TCP has difficulty fully utilizing high-speed wide-area connections. Thus, network applications are rarely able to take full advantage of high-speed networks and they are often not utilizing the available bandwidth [2].

The packet drop rate needed to fill a Gigabit pipe using the present TCP protocol is beyond the limit of currently achievable fiber optic error rates, and the congestion control becomes not so dynamic. [3]. Without expert attention from network engineers, most users are unlikely to achieve even 5 Mbps on a single stream wide-area TCP transfer, despite the fact that the underlying network infrastructure can support rates of 100 Mbps or more [4].

Researchers have worked to improve the performance of TCP in situations where there is a high bandwidth delay product and, several proposals have emerged in the literature dealing with some of the issues of this complex problem [5], [6], [7], [8], [9], [10].

Maintaining fairness among the connections in the network is an essential feature widely accepted in the community [11]. So, new solutions must coexist nicely with existing solutions, or only interfere when the existing protocols are unable to use the link capacity well.

HSTCP (HighSpeed TCP) is a recently proposed revision to the TCP congestion control mechanism. It is specifically designed for use in high-speed wide-area links. There exist few studies into the issues of its use. In this paper we report on our study into the benefits and possible deployment issues of HighSpeed TCP.

In this paper, the performance of HSTCP and the impact of its use on the present implementation of TCP is analyzed in different network conditions. These conditions include different degrees of congestion, different levels of loss rate, different degrees of bursty traffic and two distinct router queue management policies. It is expected that these different network conditions present a broad view of the strengths and weaknesses of HSTCP.

This paper is organized as follows. Section II presents an overview of the current issues faced by TCP in attempting to achieve high performance, and some of the solutions proposed to overcome these obstacles. Section III shows the foundation of HSTCP. Section IV describes the purpose of this work. Section V discusses the methodology used. The results for the experiments of this study are described in Section VI. Section VII presents a discussion of the results. Section VIII is dedicated to the conclusion.

II. TCP Performance Problems in High Speed Links

A. TCP Overview

TCP was first designed in the early 1970s. Many research, development and standardization efforts have been devoted to the TCP/IP technology since then. It is widely used in the current Internet and it is the de-facto standard transport-layer protocol.

Congestion management allows the protocol to react to and recover from congestion and operate in a state of high throughput yet sharing the link fairly with other traffic. Van Jacobson [12] proposed the original TCP congestion management algorithms. The importance of congestion management is now widely acknowledged. Many RFCs (Request For Comments)

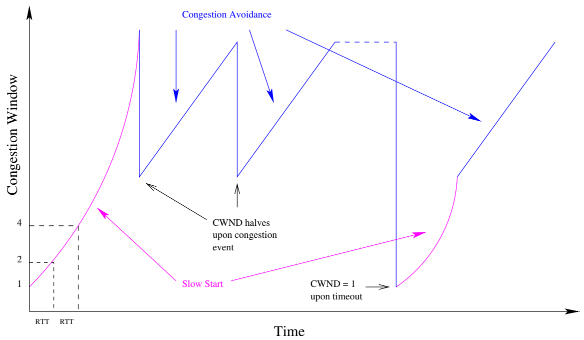


Fig. 1. TCP Congestion Control

intended to enhance TCP's performance have been published within the IETF [13], [14].

TCP's congestion management is composed of two important algorithms. The slow-start and congestion avoidance algorithms allow TCP to increase the data transmission rate without overwhelming the network. They use a variable called CWND (Congestion Window). TCP's congestion window is the size of the sliding window used by the sender. TCP cannot inject more than CWND segments of unacknowledged data into the network.

The general characteristics of the TCP algorithm are an initial relatively fast scan of the network capacity followed by a cyclic adaptive behavior that reacts quickly to congestion by reducing its sending rate, and then slowly increasing the sending rate in an attempt to stay within the area of maximal transfer efficiency. This general behavior is shown in Figure 1.

TCP's algorithms are referred to as AIMD (Additive Increase Multiplicative Decrease) and are the basis for its steady-state Congestion Control. TCP increases the congestion window by one packet per window of data acknowledged, and halves the window for every window of data containing a packet drop. The TCP congestion control can roughly be expressed in the following equations:

Congestion Avoidance

$$\textbf{ACK} : CWND \leftarrow CWND + \frac{a}{CWND} \tag{1}$$

$$\textbf{DROP} : CWND \leftarrow CWND - b \times CWND \tag{2}$$

Slow-Start

$$\textbf{ACK} : CWND \leftarrow CWND + c \tag{3}$$

The terms CWND, a and c are all defined in units of Maximum Segment Size (MSS). The canonical values for a, b and c are: a=1, b=0.5 and c=1.

B. The Problem With the TCP Congestion Avoidance Algorithm

The introduction of high-speed network technologies has opened the opportunity for a dramatic change in the achievable performance of TCP based applications. Unfortunately, this potential has not generally been realized.

The performance of a TCP connection is dependent on the network bandwidth, round trip time, and packet loss rate. At present, TCP implementations can only reach the large congestion windows necessary to fill a pipe with a high bandwidth delay product when there is an exceedingly low packet loss rate. Otherwise, random losses lead to a significant throughput deterioration when the product of the loss probability and the square of the bandwidth delay is larger then one [15]. For example, a standard TCP connection with 1500-byte packets and a 100 ms round-trip time, would require an average congestion window of 83,333 segments and a packet drop rate of at most one congestion event every 5,000,000,000 packets to achieve a steady-state throughput of 10 Gbps (this translates to at most one congestion event every 1h:40m) [16]. This loss rate is well below what is possible today with the present optical fiber and router technology.

C. Proposed Solutions for TCP Congestion Avoidance Performance Problems in HighSpeed Links

Other solutions that have been proposed to overcome this limitation include:

a) XCP: The XCP (eXplicit Control Protocol) [9] generalizes the ECN (Explicit Congestion Notification) [17], [18] proposal. Instead of the one bit congestion indication used by ECN, XCP-enabled routers inform senders of the degree of congestion at the bottleneck. Each XCP packet carries a congestion header, which is used to communicate a flow's state to routers and feedback from the routers to the receivers. Like TCP, XCP is a window-based congestion control protocol designed for best effort traffic. It decouples utilization control from fairness control. The XCP proposal claims to be stable and efficient regardless of the link capacity, the round-trip time, and the number of sources.

b) FAST TCP: FAST TCP [19] proposes that, to maintain stability, sources should scale down their responses by their individual RTT and links should scale down their response by their individual capacity, because it has been shown that the current algorithms can become unstable as delay increases, and also as network capacity increases. They claim that is possible to implement a TCP algorithm that can maintain linear stability without having to change the current link algorithm [20]. So, by modifying just the TCP kernel at the sending hosts, they can stabilize the Internet with the current routers. They implemented a FAST TCP kernel with some features: it uses both queuing delay and packet loss as signals of congestion; deals with massive losses; reduces burstiness and massive losses using pacing at sender; and converges rapidly to a neighborhood of the equilibrium value and then smoothly home in on the target.

III. HighSpeed TCP Fundamentals

A. Description

The HighSpeed TCP for Large Congestion Windows was introduced in [16] as a modification of TCP's congestion control mechanism to improve performance of TCP connections with large congestion windows.

HighSpeed TCP is designed to have a different response in environments of very low congestion event rate, and to have the standard TCP (referred to in this work as Regular TCP or REGTCP) response in environments with packet loss rates of at most 10^{-3}. Since, it leaves TCP's behavior unchanged in environments with mild to heavy congestion, it does not increase the risk of congestion collapse. In environments with very low packet loss rates (typically lower than 10^{-3}), HighSpeed TCP presents a more aggressive response function.

B. Modified Response Function

HighSpeed TCP introduces a new relation between the average congestion window w and the steady-state packet drop rate p. For simplicity, this new HighSpeed TCP response function maintains the property that the response function gives a straight line on a log-log scale (as does the response function for Regular TCP, for low to moderate congestion). Both response functions are shown in Figure 2.

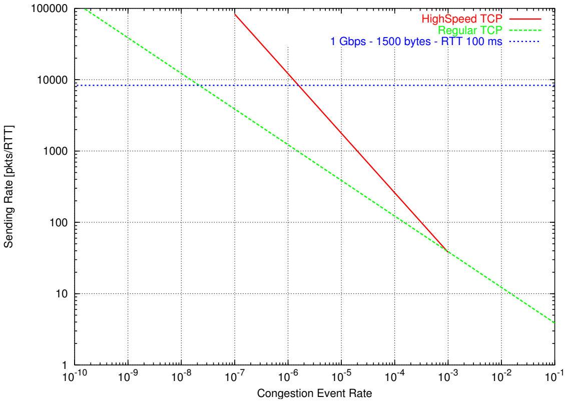


Fig. 2. HighSpeed TCP Response Function

The HighSpeed TCP response function is specified using three parameters: Low_Window, High_Window, and High_P. Low_Window is used to establish a point of transition and ensure compatibility. The HighSpeed TCP response function uses the same response function as Regular TCP when the current congestion window is at most Low_Window, and uses the HighSpeed TCP response function when the current congestion window is greater than Low_Window. High_Window and High_P are used to specify the upper end of the HighSpeed TCP response function. It is set as the specific packet drop rate High_P, needed in the HighSpeed TCP response function to achieve an average congestion window of High_Window.

The HighSpeed TCP response function is represented by new additive increase and multiplicative decrease parameters. These parameters modify both the increase and decrease parameters accordinig to CWND. In congestion avoidance phase, its behavior can be expressed in the following equations:

Congestion Avoidance

$$\textbf{ACK} \quad : \quad CWND \leftarrow CWND + \frac{a(CWND)}{CWND} \tag{4}$$

$$\textbf{DROP} \quad : \quad CWND \leftarrow CWND - b(CWND) \times CWND \tag{5}$$

IV. Investigation

The general purpose of this work was to study the effectiveness of HighSpeed TCP in high-speed long-distance links as a mechanism for bulk data transfer. Of particular concern was the study of fairness with other types of TCP already in use. More details of this study can be found in [21].

To fulfill this general objective, this study focused on specific topics as follows:

1) What is the behavior of HighSpeed TCP in situations where Regular TCP underperforms;
2) Is it possible to use HighSpeed TCP together with Regular TCP and maintain an acceptable fairness;
3) What is the effect of the router queuing policy (RED and DT) on the performance of HighSpeed TCP and on the fairness between HighSpeed TCP and Regular TCP; and
4) Can HighSpeed TCP be a substitute to other types of bulk data transfer.

The main focus was on the behavior of HighSpeed TCP and Regular TCP in situations where both were in steady-state or near to steady-state. The TCP congestion avoidance phase was of particular interest since it is where the AIMD algorithm works, and thus when the HighSpeed TCP algorithm runs. Long-lived TCP flows traveling on high-speed and long-distance links with a large amount of data to transmit were the focus. This investigation was developed using a simple topology scenario to minimize complexity and to reduce the number of variables to collect and study.

V. Methodology

The experiments were conducted using the NS-2 simulator [22].

A. Simulation Environment

1) Network Topology: The simulation network topology used was a *dumbbell* with a single bottleneck, as shown in Figure 3. All traffic passed through the bottleneck link. The bottleneck link bandwidth was 1 Gbits/s, the link delay was 50 ms. The simulations used two types of router queue management, DT (DropTail) and RED (Random Early Detection) [23]. In the case of RED, ECN was also used. The queue size at each router was the bandwidth delay product in packets.

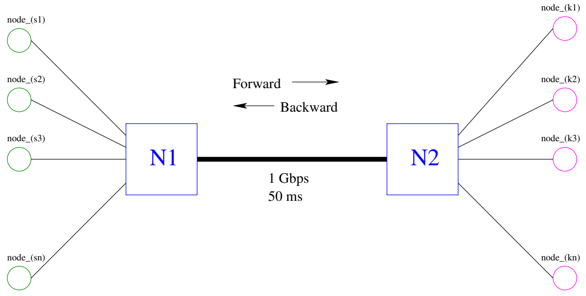


Fig. 3. Network Topology

2) TCP Flows Setup: The TCP flows had the ECN bit set; the packet size was 1500 bytes; the maximum window size was large enough to not impose limits; random times between sends were set to avoid phase effects [24]; the flows used a modified version of the Limited Slow-Start algorithm for large congestion windows [5].

The TCP agent used for the sender and the receiver was SACK [25]. FTP was the application used to transmit data through the TCP connections. All the HSTCP flows used the forward direction. The HSTCP parameters used are given in Table I.

TABLE I

HSTCP PARAMETERS

HSTCP Parameters	Value
Low_Window_	31
High_Window_	83000
High_P_	0.0000001

For comparison, the HSTCP flows were run against TCP SACK implementations. A set of web-like flows and a set of small TCP flows were also used as *background noise* for all the simulation. Other TCP flows were used to represent bursty traffic. They were short-lived flows that lasted for a few seconds.

3) Data Collection Configuration: The aggregated data was collected in two halves. Only the second half was of interest because this research focused on the steady-state behavior. Each experiment was run ten times, for three hundred seconds. The line shown is the median of these simulations.

B. Descriptions of Scenarios for the Experiments

We used three sets of primary flows in most of this study. The first set had only HSTCP flows, the second was composed only of REGTCP flows, and the third set contained a combination of REGTCP and HSTCP flows. The number of flows for each set varied according to the experiment.

The three sets of flows were each exposed to different network environments. In the first network environment there were no other traffic sources and no extra interference beyond that generated by the REGTCP and HSTCP flows. This network environment we refer to as *Ideal Condition*. The second network environment represented the situation where there were systemic losses (or losses not directly related to congestion). We call it *Lossy Link Condition*. Some number of packets were randomly dropped from the flows, with a defined drop rate. The third network environment explored the reaction of the flow sets to bursty traffic, so we refer to it as *Bursty Traffic Condition*. The bursty traffic was composed of short-lived standard TCP flows running for a few seconds.

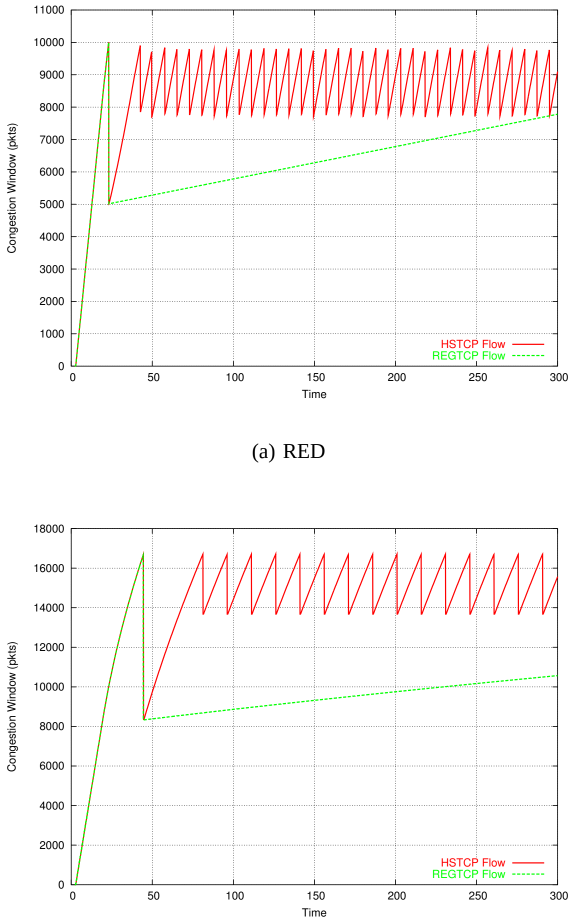
VI. RESULTS FROM THE EXPERIMENTS

This section presents the results of our experiments. All experiments were run with both RED and DT queuing policy. In the cases where there is a significant difference in the results, results for each individual queuing policy are presented.

A. Isolated Flows

This first experiment allowed us to observe the basic behavior of a single REGTCP or a single HSTCP flow run in isolation. The experiment ran only one time, without external interference.

As shown by Figure 4, a REGTCP flow has a slower growth compared to the HSTCP flow. The REGTCP flow takes around 300 seconds to reach the bandwidth limit in congestion avoidance. In comparison, the HSTCP flow reaches this point before 50 seconds (RED case). The second observation is that HSTCP has an oscillatory behavior with a very short period. The third important observation is the influence of router queue management on the behavior of both flows. With DT queuing, drops did not occur until the router buffer overflowed. With RED queuing, the router sends the congestion signal earlier. In the case of an empty network this can lead to lower utilization with RED.



(a) RED

(b) DT

Fig. 4. Evolution of Congestion Window for a Single Flow

B. Ideal Condition

This set of experiments provided a baseline by showing the behavior of a variable number of REGTCP and HSTCP flows, when there was no external interference, except the background traffic. Figure 5 shows the link utilization, for the REGTCP and HSTCP flows.

This graphic shows that the HSTCP flows can reach full link utilization with a small number of flows. REGTCP needs a higher number of flows to approach 100% link utilization.

The following graphic in Figure 6 presents the congestion event rate for the first and second set of flows, when RED was

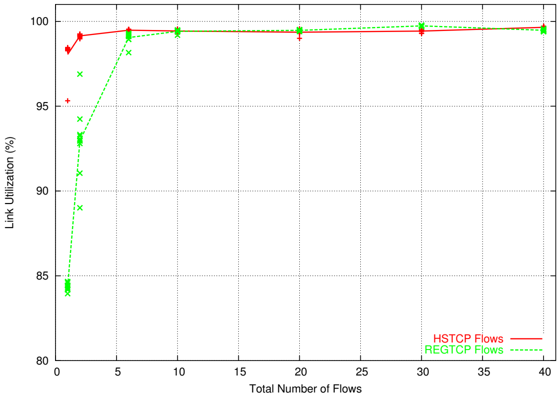


Fig. 5. Link Utilization - Ideal Condition - RED

deployed.

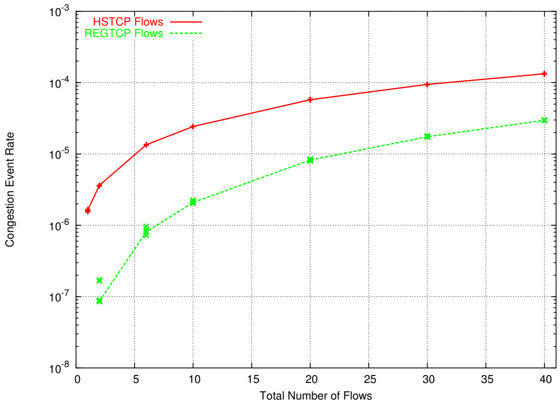


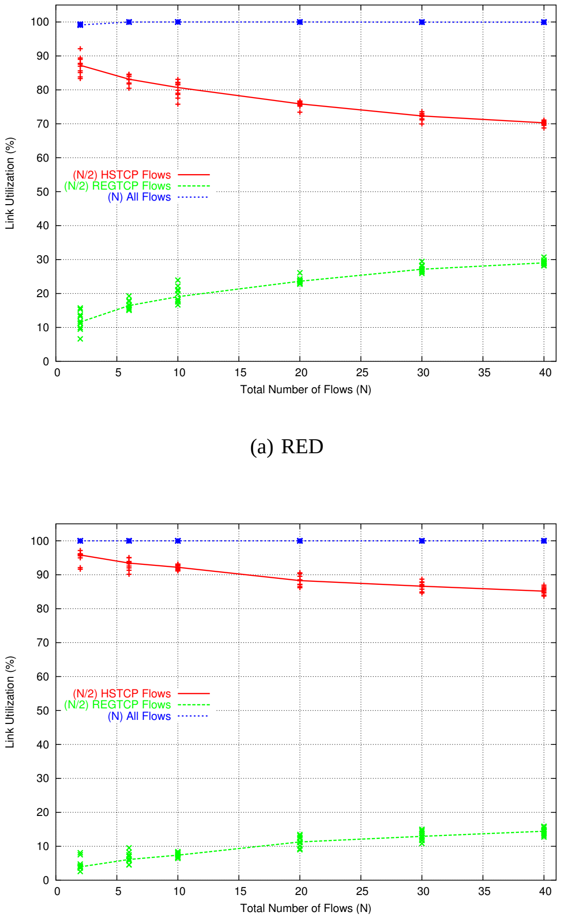
Fig. 6. Congestion Event Rate - Ideal Condition - RED

These graphics show that there is a clear difference between the congestion event rate resulting from HSTCP and REGTCP flows. HSTCP produces a higher congestion event rate. Another important aspect to observe is that the congestion event rate for HSTCP is never lower that 10^{-6}, as expected from the HSTCP parameters. When we used DT as the router queue management policy, it generated a slightly higher rate of congestion events but otherwise was similar.

The link utilization achieved by the set of mixed flows is presented in the graphs of Figure 7. The performance is separated by flow type; one line is the aggregated result of the HSTCP flows and the other is the aggregated result of the REGTCP flows. The third line is the result for all the flows combined.

These graphs show that when HSTCP flows are directly competing with REGTCP flows, the bandwidth share used by HSTCP is higher than the bandwidth used by the REGTCP flows. This fact is independent of the type of router queue management used. Also, the bandwidth share used by HSTCP decreases as the total number of flows increase.

The relative fairness for the mixed flow set is depicted in Figure 8. It shows the ratio between the amount of bandwidth used by all the HSTCP flows and the amount of bandwidth used by all the remaining REGTCP flows.



(a) RED

(b) DT

Fig. 7. Link Utilization - Ideal Condition - Mixed Flows

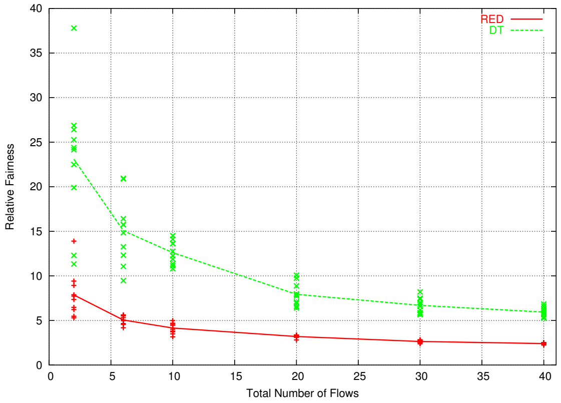


Fig. 8. Relative Fairness - Ideal Condition

This graph reveals that the fairness improves as the number of flows increases. It is also important to observe that when RED is deployed, the relative fairness is better than when DT is used.

The last result presented in Figure 9 is the amount of aggregated bandwidth stolen from all the REGTCP flows when they are deployed together with HSTCP flows. This result is calculated using the difference between the link utilization achieved by a number of REGTCP flows when they are competing against M other REGTCP flows, and the link utilization achieved by the same number of REGTCP flows when they are competing against M other HSTCP flows.

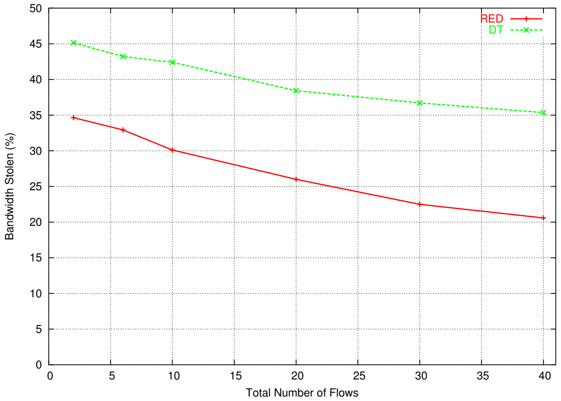


Fig. 9. Bandwidth Stolen - Ideal Condition

This graph shows that the amount of bandwidth stolen decreases as the number of flows increases. This fact highlights that the HSTCP aggressiveness adapts as the traffic condition changes. Although the amount of bandwidth stolen decreases as the number of flows increase, the distance between the amounts stolen between RED and DT increases slightly

C. Lossy Link Condition

The focus of this set of experiments was to observe the behavior of REGTCP and HSTCP flows when subjected to systemic losses. We used the simulator error model to simulate losses on the bottleneck link. This loss model was set to drop a packet with a defined average drop rate. We used three sets of flows to develop this experiment. The first set contained 10 HSTCP flows, the second set contained 10 REGTCP flows and the third one contained a mix of 5 HSTCP and 5 REGTCP flows.

Figure 10 presents the performance of the link utilization metric, for the first and second set of flows.

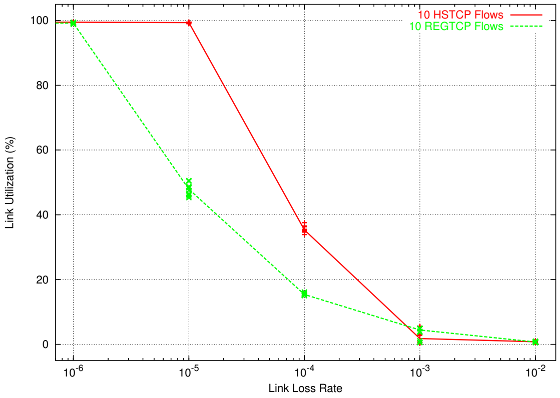


Fig. 10. Link Utilization - Lossy Link Condition - RED

The set of REGTCP flows show a rapid performance loss as the link loss rate increases. In contrast, the HSTCP flows showed better resilience to moderate link loss, and consistently used more bandwidth than the REGTCP flows.

The link utilization achieved by the mixed set of flows is presented in Figure 11. Here the performance is separated by flow type; one line is the aggregate result of the HSTCP flows and other line is the aggregated result of the REGTCP flows. The third line is the result of all the flows combined.

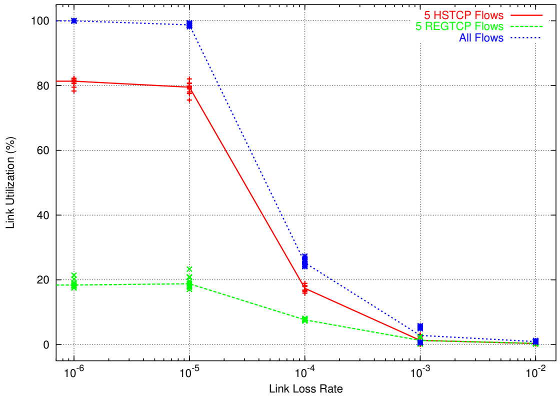


Fig. 11. Link Utilization - Lossy Link Condition - Mixed Flows - RED

We see that, as expected, the difference between the bandwidth used by the HSTCP flows and that used by the REGTCP flows decreases as the number of losses increases. Another important aspect to point out is that, for a link loss rate around 10^{-5}, the link is fully utilized, and below this rate, congestive losses will be dominant.

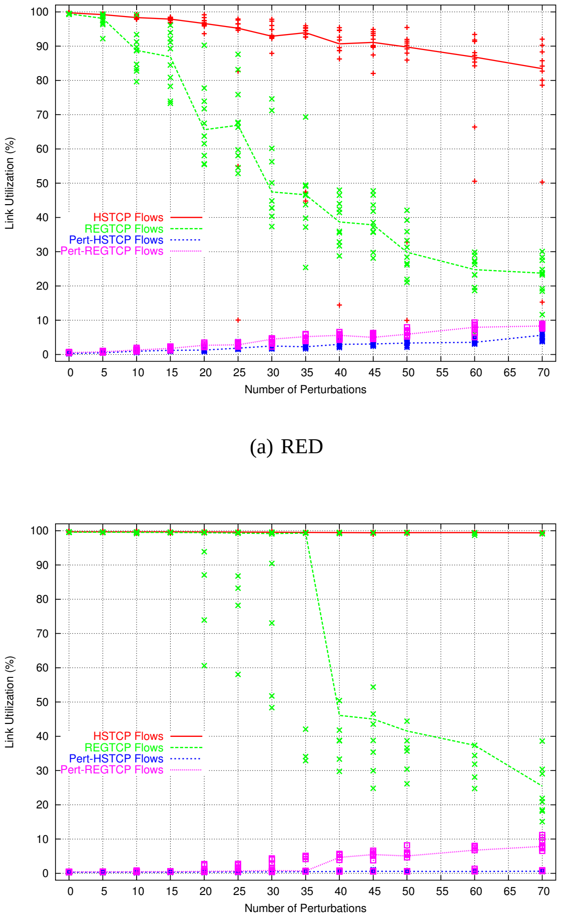
D. Bursty Traffic Condition

This set of experiments helps us to understand the reaction of the REGTCP and the HSTCP flows when they are subjected to bursty traffic. We introduced perturbation in the form of bursty flows randomly distributed throughout the simulation time. We used three sets of flows in this experiment. The first set contained 10 HSTCP flows, the second set contained 10 REGTCP flows and the third one contained a mix of 5 HSTCP and 5 REGTCP flows.

The graphs in Figure 12 present the performance of the link utilization metric, for the REGTCP and HSTCP sets of flows It also shows the link utilization of the perturbing flows present.

We observe from Figure 12(a) that the set of HSTCP flows decreases their link utilization smoothly and slowly as the number of perturbations increase. On the other hand, the impact on the set of REGTCP flows is higher, and their performance goes down quickly as the number of perturbations increases.

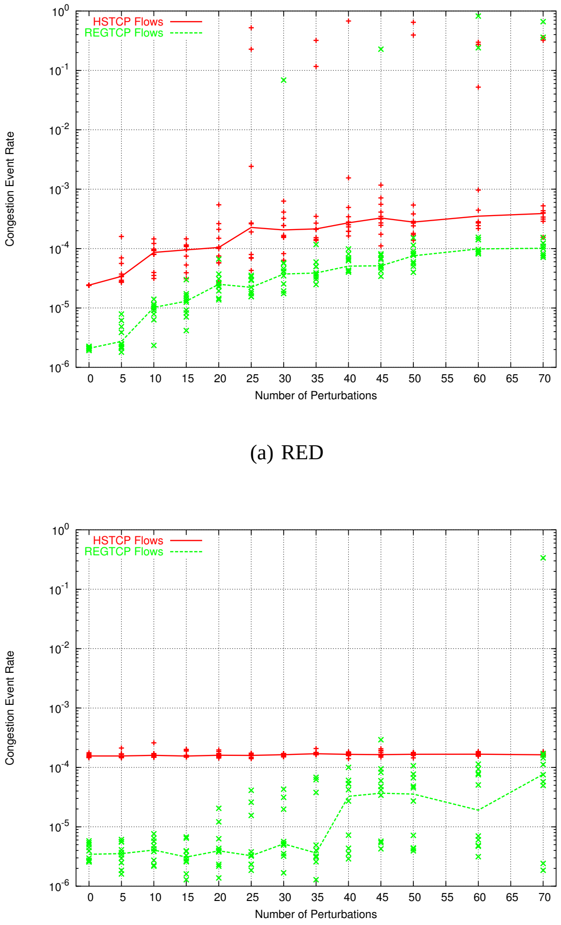
The impact of the use of distinct router queuing management is clear when the set of HSTCP flows is submitted to bursty traffic. The link utilization with HSTCP flows decreases slightly with RED, but it is almost immune to the perturbations when the DT router queuing policy is used, as can be seen in Figure 12(b).



(a) RED

(b) DT

Fig. 12. Link Utilization - Bursty Traffic Condition



(a) RED

(b) DT

Fig. 13. Congestion Event Rate - Bursty Traffic Condition

Figure 13 presents the congestion event rate for the REGTCP and HSTCP sets of flows, when RED and DT are deployed.

We observe that the congestion event rate increases continuously as the number of perturbations increases, when RED router queue management is used. When DT router queue management is deployed, this behavior changes. The set of HSTCP flows presents an almost constant congestion event rate, and the set of REGTCP flows has two levels of congestion event rates. It could be caused by the occurrence of global synchronization [26] or by the increased burtiness from REGTCP flows slow-stating.

The link utilization achieved by the mixed set of flows is presented in Figure 14. One line is the aggregated result of the HSTCP flows, and other line is the aggregated result for the REGTCP flows. The third line is the result of all the flows combined. The remaining line represents the link utilization of the perturbations.

The important information provided by these graphics is the poor utilization of the set of REGTCP flows. However, this performance remains relatively constant as the number of perturbations increases.

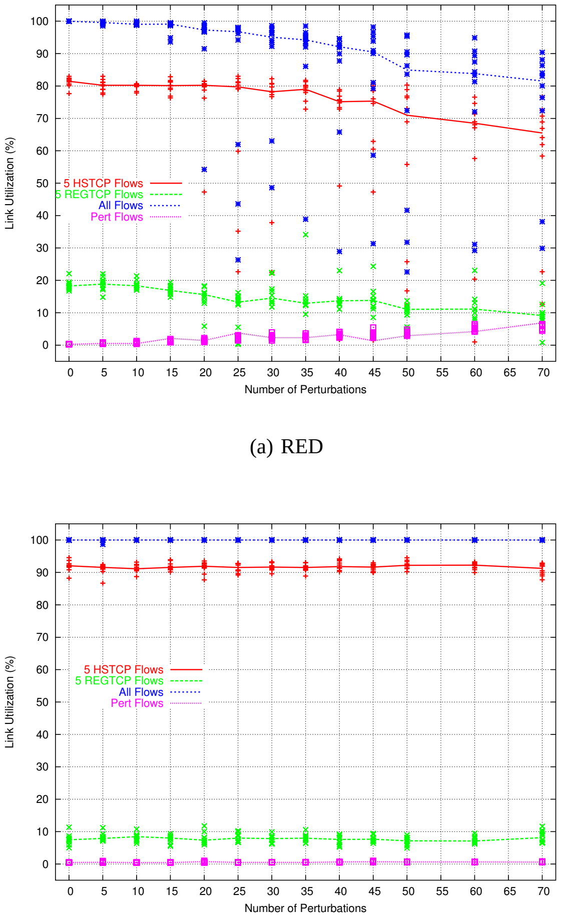
The relative fairness for the mixed flow set is depicted in Figure 15. It is almost constant for both queuing policies. The HSTCP flows get between 10 and 15 times more bandwidth share than the REGTCP flows, for DT, and they get around 5 times more, when RED is used.

E. Competition among Heterogeneous Flows

This set of simulations was to verify the behavior of a HSTCP flow when a varying number of REGTCP flows were deployed with it.

The graphs in Figure 16 present the performance of the link utilization metric when RED and DT router queue management are used, respectively. They also present the link utilization of all the flows together.

Some interesting results are represented in these graphs. The first is that the HSTCP flow adapts with the amount of REGTCP flows used, and it avoids allowing the link to become idle. The second happens at the crosspoint of the HSTCP line and the REGTCP line. This shows the number of REGTCP flows that have equivalent performance to 1 HSTCP flow. This



(a) RED

(b) DT

Fig. 14. Link Utilization - Bursty Traffic Condition - Mixed Flows

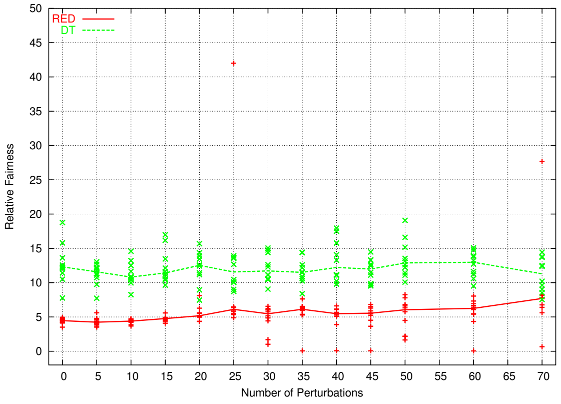
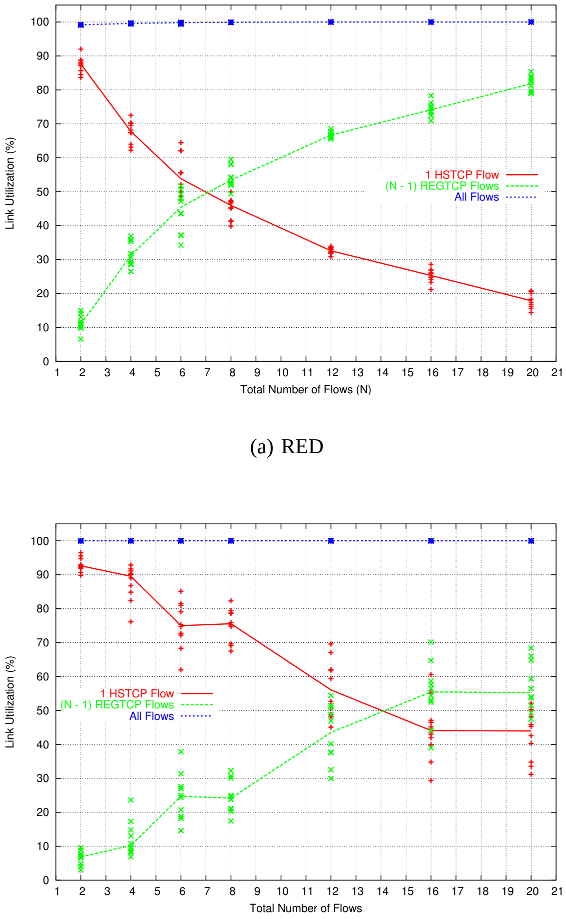


Fig. 15. Relative Fairness - Bursty Traffic Condition

number appears to be highly dependent on the type of router queue management used.

The ratio between the amount of bandwidth used by the HSTCP flow and the amount of bandwidth used by the combined REGTCP flows is depicted in Figure 17.



(a) RED

(b) DT

Fig. 16. Link Utilization - Competition Among Heterogeneous Flows

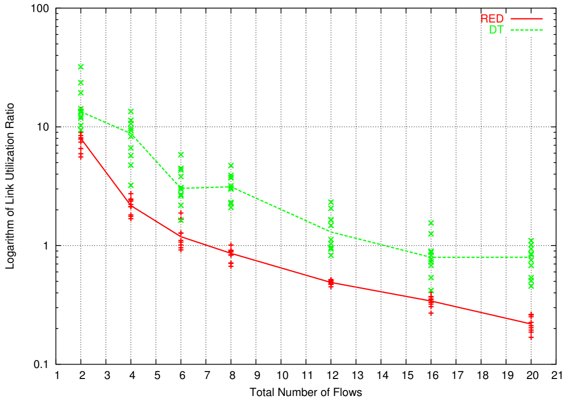


Fig. 17. Link Utilization Ratio - Competition Among Heterogeneous Flows

F. Constant Link Loss of 10e-5

This set of experiments repeats the Ideal Condition experiment, except that it introduces a constant link loss rate of 10^{-5}. The purpose of this change is to investigate the behavior of the HSTCP and the REGTCP flows with systemic losses and a variable number of flows for each set of flows.

Figure 18 presents the performance of the link utilization metric, for the REGTCP and HSTCP sets of flows.

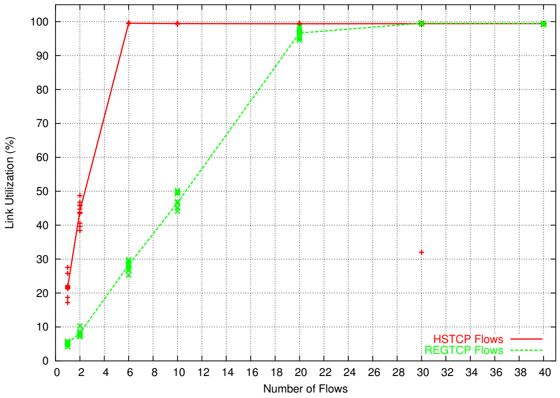
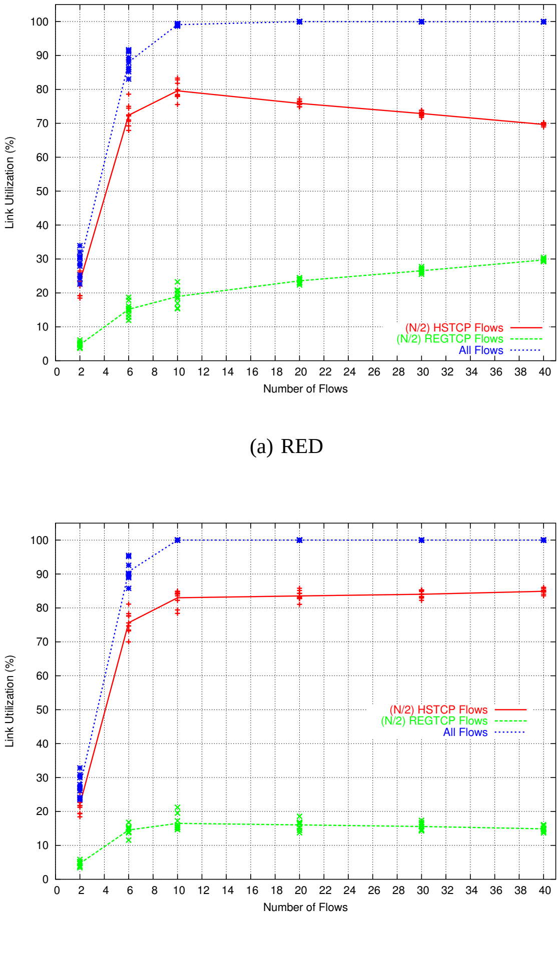


Fig. 18. Link Utilization - Constant Link Loss Rate of 10^{-5} - RED

This graph shows that when HSTCP flows are deployed in this network condition, there is need of 6 flows to reach full link utilization. But, when REGTCP flows are used, this number increases to 20 or more.



(a) RED

(b) DT

Fig. 19. Link Utilization - Constant Link Loss Rate of 10^{-5} - Mixed Flows

The link utilization achieved by the mixed set of flows is presented in Figure 19. The performance is presented by flow type.

These graphs show the influence router queue management has on the behavior of link utilization for each type of flow. While for RED, the link utilization for the HSTCP flows decreases as the total number of flows increases, for DT, the same link utilization stays constant or even slightly increases, as the number of flows increase.

G. *Parallel Streams on Lossy Link Condition*

The focus of this set of experiments was to observe the impact of Parallel Streams on long-lived REGTCP flows and to compare it with the impact of HSTCP over the same long-lived REGTCP flows, when both are subjected to systemic losses. We used two sets of flows to develop this experiment. The first set contained 10 REGTCP flows (representing the long-lived flows) and 1, 4, 7, 10, 20 or 30 parallel REGTCP streams. The second set is formed by the same 10 REGTCP flows of the first set and one HSTCP flow.

We present here only the per flow relative fairness. The intention is to show the competition that a parallel stream transmission represents for a single long-lived regular TCP flow. The amount of link bandwidth used for the aggregate parallel stream transmission is divided by the amount of link bandwidth used by one of the 10 long-lived streams. The same procedure is used for the case of the transmission using one HSTCP flow. The results are presented in Figure 20.

It is clear that, when parallel streams are deployed, the relative fairness is almost constant over a wide range of link loss rates. This behavior only changes when there is a heavy packet loss rate. In contrast, the relative fairness when HSTCP is used is not constant and has a wide range of values.

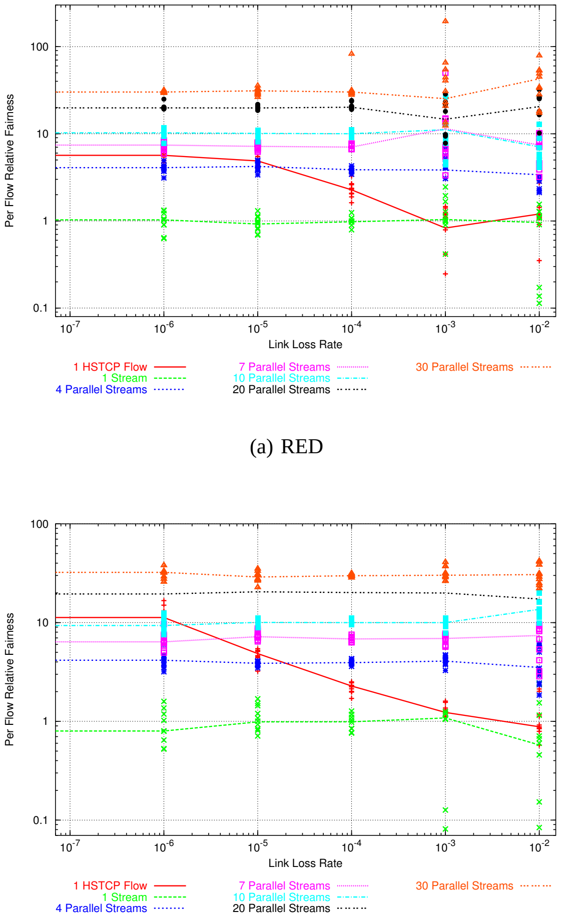
H. *Parallel Streams on Bursty Traffic Condition*

The objective of this last set of experiments was to observe the impact of Parallel Streams on long-lived REGTCP flows and to compare it with the impact of HSTCP over the same long-lived REGTCP flows when they are subjected to bursty traffic.

We used two sets of flows to develop this experiment. The first set contains 10 REGTCP flows (representing the long-lived flows) and also 1, 4, 7, 10 or 20 parallel streams. The second set is formed by the same 10 REGTCP flows of the first set and one HSTCP flow.

We present the results of per flow relative fairness in Figure 21. We observe in these graphics that when RED is used the relative fairness increases as the number of perturbations increase, but this behavior is not clear when DT is deployed. In the DT case the ratio between the bandwidth used by HSTCP and the bandwidth used by the 10 long-lived flows spreads over a wide range of values.

Figure 22 shows the performance just of the parallel streams and the HSTCP flow in this context. The HSTCP flow improved its performance as the number of perturbations increased (RED case) until around 40 perturbations, but this improvement didn't happen by stealing bandwidth from the



(a) RED

(b) DT

Fig. 20. Per Flow Relative Fairness - Parallel Streams on Lossy Link Condition

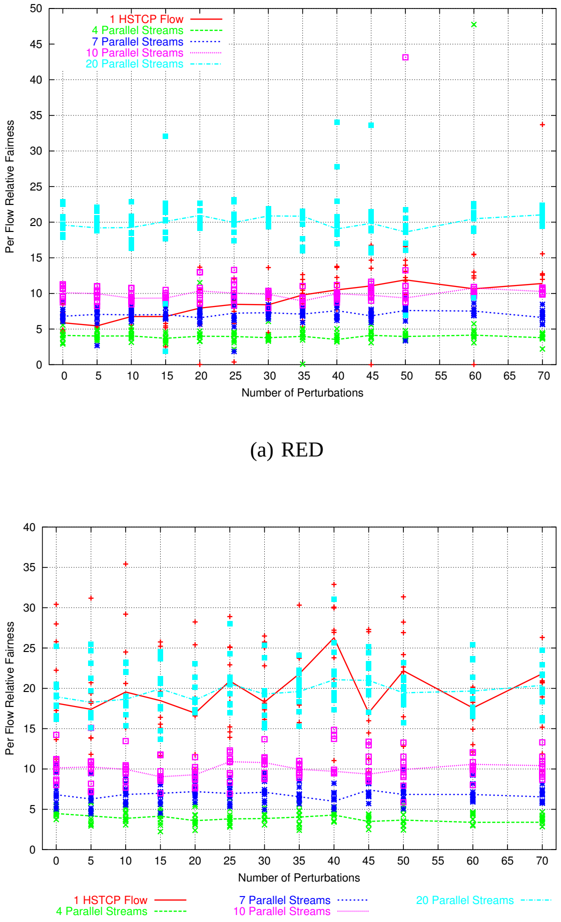
10 long-lived flows, as seen on Figure 23. The HSTCP is able to use bandwidth share the REGTCP flows and the parallel streams are unable to use in this situation.

VII. Issues For The Deployment of HighSpeed TCP

There are several relevant aspects to consider when deploying HighSpeed TCP. These are grouped below by topic.

A. Comparison with Regular TCP

As was established in the previous section, HighSpeed TCP performs better than Regular TCP for high-speed long-distance links. The major drawback in Regular TCP is that it dramatically reduces the size of the congestion window in response to a congestion event and its ACK-clocked congestion window grows in increments of one. This leads to slow recovery from a congestion event when the congestion window was very large and leaves the link with a low level of utilization during significant periods of time. In comparison, HighSpeed TCP cuts the window less and it grows faster so its recovery takes less time. This characteristic increases its average link utilization.



(a) RED

(b) DT

Fig. 21. Per Flow Relative Fairness - Parallel Streams on Bursty Traffic Condition

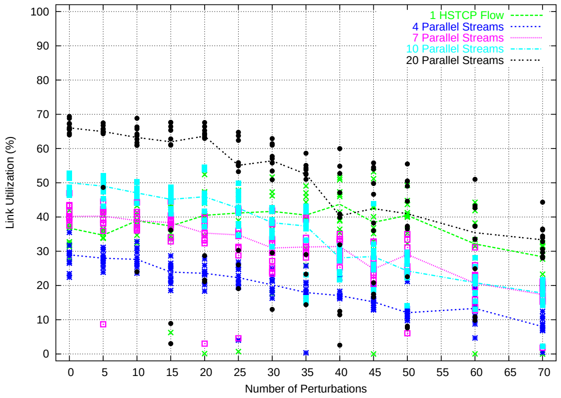


Fig. 22. Aggregated Link Utilization of Competing Parallel Streams - Parallel Streams on Bursty Traffic Condition - RED

In the presence of systemic losses, Regular TCP flows show poor link utilization. For the conditions of our study, a link loss rate between 10^{-5} and 10^{-4} prevented Regular TCP from making reasonable use of the link bandwidth available (less than 50% in our case). In this range, the HighSpeed TCP flows

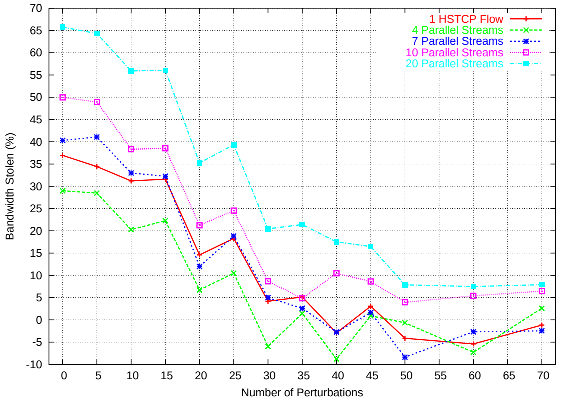


Fig. 23. Aggregated Bandwidth Stolen - Parallel Streams on Bursty Traffic Condition - RED

were able to use almost double the bandwidth used by Regular TCP.

The Regular TCP flows also experienced low link utilization in the presence of bursty traffic. The bursty traffic was using less than a 10% share of the link bandwidth but the Regular TCP flows dropped their bandwidth by around 70%. This was due to the bursty nature of these perturbations. The HSTCP flows decreased their link utilization also, but not as dramatically.

B. Fairness Impact

The bandwidth share used by the HighSpeed TCP flows was higher than that used by Regular TCP flows, when both types of flows competed for the same link. However, it was noticeable that the amount of the link bandwidth used by the HighSpeed TCP flows decreased as the total number of flows increased. The opposite happened with the Regular TCP flows. The reason for this behavior was that the higher the number of flows competing for the link bandwidth, the more congestion events happened and the less aggressive the HighSpeed TCP flows became. With the decrease in the HighSpeed TCP aggressiveness, the Regular TCP flows had more opportunity to use the bandwidth available.

This result highlights two distinct characteristics of the HighSpeed TCP protocol. It is more aggressive at using the available bandwidth, but it decreases its aggressiveness as the congestion event rate increases. This adaptability is very interesting in the context of high speed links. It avoids having a link become idle caused by the slow dynamic of Regular TCP, and yet it does not prevent more Regular TCP streams from obtaining a reasonable share of the link.

Bursty traffic had only a small influence on the amount of bandwidth that the HighSpeed TCP flows stole from the Regular TCP flows, thus it had little influence on the fairness.

C. Effects of Router Queue Management

The change of the queue management scheme did not significantly affect the link utilization of HighSpeed TCP flows in most cases. It did, however, cause a difference in the amount of congestion events; RED requires fewer congestion events to control the TCP sending rate, compared to DT.

The impact of the use of different router queue management policies was clear when HighSpeed TCP was submitted to bursty traffic. The link utilization for HighSpeed TCP flows decreased slightly with RED, but it was not affected when DT router queue policy was used.

The general pattern of the aggregated relative fairness found, when RED was used, was the same when DT router queue management was deployed. The difference was the higher amount of bandwidth that HighSpeed TCP flows took from Regular TCP flows. When they were submitted to bursty traffic condition, the aggregated relative fairness was almost constant using DT, but seemed to increase slightly when RED was used.

There was also a difference in the number of equivalent Regular TCP flows for one HighSpeed TCP flow, when both were competing for the same bandwidth and different router queue management were deployed. In this work six Regular TCP flows equaled one HighSpeed TCP flow, with RED, and thirteen Regular TCP flows equaled one HighSpeed TCP flow, in the case of DT. But these numbers depend totally on the average packet drop rate in the experiment.

D. Use For Bulk Data Transfer

Deployment of HighSpeed TCP requires changes to the TCP stack at the TCP data sender, but once these changes are made all applications can benefit from them. This represents an advantage over other types of bulk data transfer such as parallel streams. For parallel streams it is necessary to change the application programs and to know a priori the number of parallel flows to transmit. Also, in our opinion, HighSpeed TCP presents better fairness and adaptability to an environment of variable congestion event rates than other bulk data transfer mechanisms, because of its different response function. Parallel streams may also present better adaptability if they use some kind of adaptative control, as fractional congestion control [27], but at a cost of their simplicity.

If HighSpeed TCP is used on a network with significant systemic packet losses, the packet loss rate will define the maximum available throughput. It is possible to change this limit by using different values for the HSTCP parameters of Low_Window, High_Window, and High_P. In all situations, HighSpeed TCP should, by design, perform better or the same as Regular TCP.

VIII. Conclusion

TCP has difficulty fully utilizing network links with a high bandwidth delay product. HighSpeed TCP outperforms TCP in this condition, and has an adaptability that makes an incremental adaption approach easy. HighSpeed TCP is easy to deploy avoiding changes in routers and programs.

HighSpeed TCP is appropriate to bulk data transfer applications, because it is able to maintain high throughput in different network conditions, and it is easy to deploy when compared with other solutions already in use.

A point of concern is its fairness at low speeds, mainly in networks with droptail routers. A better relation with TCP may be achieved by adjusting its three parameters, in particular Low_Window. At high speeds it is not possible to maintain a fair relationship, nor is it desirable, using the present concept of fairness. Pacing could help with droptail routers, and improve fairness. Futher studies in this area should be carried out.

Investigations in real network are necessary to complete the assessment of its deployment. Some have already began [28], [29]. So far, no unexpected behavior was found. HighSpeed TCP behaved as forseen by its response function, and appears to be a real and viable option for use on high-speed wide area TCP connections.

ACKNOWLEDGEMENT

We would like to thank Sally Floyd for many fruitful discussions, insights, and constant technical support during the development of this study.

This work was supported by the Director, Office of Science, Office of Advanced Computing Research, Mathematical Information and Computing Sciences Division, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. See the disclaimer at http://www-library.lbl.gov/disclaimer.

REFERENCES

[1] M. Fisk and W. Feng, “Dynamic right-sizing in TCP,” in *Procedings of Los Alamos Computer Science Institute Symposium*, October 2001.

[2] W. Huntoon, T. Dunigan, and B. Tierney, “The Net100 project: Development of network-aware operating systems,” URL http://www.net100.org.

[3] S. Floyd, S. Ratnasamy, and S. Shenker, “Modifying TCP’s congestion control for high speeds,” 2002, Preliminary Draft. URL http://www.icir.org/floyd/papers/hstcp.pdf.

[4] B. Irwin and M. Mathis, “Web100: Facilitating high-performance network use,” January 2001, URL http://www.internet2.edu/E2E/papers/20010109-E2EPM-Irwin.pdf.

[5] S. Floyd, “Limited slow-start for TCP with large congestion windows,” May 2002, internet draft draft-floyd-tcp-slowstart-00b.txt, work in progress.

[6] J. Lee, D. Gunter, B. Tierney, W. Allock, J. Bester, J. Bresnahan, and S. Teck, “Applied techniques for high bandwidth data transfers across wide area network,” in *Computing in High Energy and Nuclear Physics*, Beijing, China, April 2001, LBNL-46269.

[7] M. Sooriyabandara and G. Fairhurst, “Performance limitations due to TCP burstiness in GEO satellite networks with limited buffering,” in *London Communications Symposium*, London, UK, September 2000, pp. 127–130.

[8] J. Semke, J. Mahdavi, and M. Mathis, “Automatic TCP buffer tuning,” *Computer Communication Review*, vol. 28, no. 4, pp. 315–323, October 1998.

[9] D. Katabi, M. Handley, and C. Rohrs, “Internet congestion control for high bandwidth-delay product networks,” in *ACM Sigcomm 2002*, August 2002.

[10] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge, “Paced TCP for high delay-bandwidth networks,” in *Proceedings of IEEE Globecom*, December 1999.

[11] G. Hasegawa and M. Murata, “Survey on fairness issues in TCP congestion control mechanisms,” *IEICE Transactions on Communications*, vol. E84-B, no. 6, pp. 1461–1472, June 2001.

[12] V. Jacobson, “Congestion avoidance and control,” in *Proceedings of the ACM SIGCOMM ’88 Conference on Communications Architectures and Protocols*, vol. 18, Stanford, CA, August 1988, pp. 314–329.

[13] R. Braden, “Requirements for internet hosts - communication layers,” Internet Engineering Task Force, October 1989, RFC1122.

[14] V. Jacobson, R. Braden, and D. Borman, “TCP extensions for high performance,” Internet Engineering Task Force, May 1992, RFC1323.

[15] T. Lakshman and U. Madhow, “Performance analysis of window-based flow control using TCP/IP: Effect of high bandwidth-delay products and random loss,” in *Fifth International Conference on High Performance Networking*, June 1994, pp. 135–149.

[16] S. Floyd, “Highspeed TCP for large congestion window,” February 2003, Internet Draft draft-floyd-tcp-highspeed-01.txt, work in progress.

[17] K. Ramakrishnan and S. Floyd, “A proposal to add explicit congestion notification (ECN) to IP,” Internet Engineering Task Force, January 1999, RFC2481.

[18] K. Ramakrishnan, S. Floyd, and D. Black, “The addition of explicit congestion notification (ECN) to IP,” Internet Engineering Task Force, September 2001, RFC3168.

[19] C. Jinand, D. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, and S. Singh, “FAST TCP: From theory to experiments,” April 2003, submitted to IEEE Comunications Magazine, Internet Technology Series.

[20] H. Choe and S. Low, “Stabilized Vegas,” in *IEEE Infocom*, San Francisco, CA, April 2003.

[21] E. Souza, “A simulation-based study of HighSpeed TCP and its deployment,” Lawrence Berkeley National Laboratory, Tech. Rep. LBNL-52549, April 2003.

[22] T. V. Project, “NS-2 network simulator,” URL http://www.isi.edu/nsnam/ns.

[23] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, “Recommendations on queue management and congestion avoidance in the Internet,” Internet Engineering Task Force, April 1998, RFC2309.

[24] S. Floyd and V. Jacobson, “On traffic phase effects in packet-switched gateways,” *Internetworking: Research and Experience*, vol. 3, no. 3, pp. 115–156, September 1992.

[25] K. Fall and S. Floyd, “Simulation-based comparisons of Tahoe, Reno and SACK TCP,” *Computer Communication Review*, vol. 26, no. 3, pp. 5–21, 1996.

[26] L. Zhang, S. Shenker, and D. Clark, “Observations on the dynamics of a congestion control algorithm: The effects of two way traffic,” in *Proceedings of ACM SIGCOMM’91*, Zurich, Switzerland, September 1991, pp. 133–148.

[27] T. J. Hacker, B. D. Noble, and B. D. Athey, “The effects of systemic packet loss on aggregate TCP flows,” in *IEEE/ACM Supercomputing 2002: High Performance Networking and Computing*, November 2002.

[28] F. Coccetti and L. Cottrell, “TCP stacks comparison with a single stream,” Stanford Linear Accelerator Center, March 2003, URL http://www-iepm.slac.stanford.edu/monitoring/bulk/fast/tcp-comparison.html.

[29] A. Antony, J. Blom, C. de Laat, J. Lee, and W. Sjouw, “Microscopic examination of TCP flows over transatlantic links,” January 2003, preprint submitted to Elsevier Science.